**CLAIMS**

What is claimed is:

1. A method for achieving client to server end to end security guarantees, the method comprising:

employing a proxy between the client and the server to provide connection links between the client and the server;

embedding a secure coprocessor for use as an agent of the client and/or the server;

the coprocessor acting as a converter between at least one protocol the client supports and at least one other protocol supported by the server; and

employing respective security protocols of said at least one protocol and said at least one other protocol.

2. A method as recited in claim 1, wherein the coprocessor is located at the site of the proxy.

3. A method as recited in claim 1, further comprising the step of the coprocessor guaranteeing that an application embedded in the coprocessor performs to a degree of security proscribed by the client and/or server.

4. A method as recited in claim 1, further comprising the coprocessor assuring that the proxy can not tamper with the functioning of the agent.

5. A method as recited in claim 1, wherein the client is a pervasive computing device.

6. A method as recited in claim 1, further comprising the step of adapting content supplied by the client to fit constraints of the server and/or the connection links.

7. A method for providing secure communications on a network, the method comprising;

securely embedding an agent at a proxy in the network, and

splicing a plurality of secure communication protocols of different protocol suites into the agent.

8. A method as recited in claim 7, wherein the step of splicing includes splicing a security protocol of the Wireless Applications Protocol suite (WAP) to that of the Internet protocol (IP) suite.

9. A method as recited in claim 8, wherein the Wireless Applications Protocol suite is used by a pervasive computing device.

10. A method as recited in claim 7, further comprising the agent performing at least one content adaptation function.

11. A method as recited in claim 10, wherein the step of performing includes maintaining communication privacy.

12. A method as recited in claim 7, further comprising maintaining a state of a splicing process resulting from the step of splicing.

13. A method as recited in claim 12, wherein the step of maintaining includes employing a storage device external to the proxy, and using cryptographic means to encrypt the state.

14. A method for providing network security to a network employing a proxy, the method comprising:

embedding a trusted application in a secure coprocessor located at the site of a proxy; and

delegating to a networking infrastructure a task of enforcing a trust model.

15. A method as recited in claim 14, further comprising guaranteeing that the application is trusted

to enforce the trust model between at least one server and a plurality of clients.

16. A method as recited in claim 14, further comprising assuring the tamper resistance of the application.

17. A method for secure communication between a client and a server employing an untrusted proxy; the method comprising:

embedding a coprocessor at the proxy;

the proxy receiving a specific communication request from a client;

the proxy forming an n-tuple for the specific communication;

the proxy forwarding the n-tuple to the coprocessor;

the coprocessor generating a response, including a directive, to the n-tuple;

the coprocessor sending the response to the proxy; and

the proxy implementing the directive.

18. A method of claim 17, wherein the coprocessor is a secure coprocessor.

19. A method of claim 17, wherein the step of receiving includes:

awaiting a connection request from a client;

creating an entry in a storage module for the client;

determining a sender of each received packet; and

retrieving a stored entry.

20. A method of claim 19, wherein the n-tuple includes a sender id, an entry from a storage module and the received packet.

21. A method of claim 17, wherein the client and the server can be either a sender or a receiver, and the step of generating includes employing a first protocol from the sender to the proxy and a second protocol from the proxy to the receiver and translating between the said first and second protocols.

22. A method of claim 21, wherein the translating includes decrypting the received packet as specified by the security parameters negotiated as per the first protocol and encrypting the decrypted packet as

specified by the security parameters of the second protocol.

23. A method of claim 21, wherein the translating includes modifying the received packet to meet constraints of the receiver and wherein the directive includes forwarding to the receiver the packet resulting from the step of modifying.

24. A method as recited in claim 23, further comprising aggregating a plurality of packets into a group of packets and performing content adaptation on the group of packets.

25. A method of claim 17, wherein the communication between the client and the proxy employ protocols specified by the Wireless Application Protocol suite [WAP].

26. A system to control security of a proxy interconnecting a client to a server, the system comprising:

a secure coprocessor used as an agent of the client and/or the server; and

an application embedded in the coprocessor which acts as a converter between at least one protocol the client supports and at least other protocol supported by the server, wherein the secure coprocessor employs

respective security protocols of said at least one protocol and said at least one other protocol.

27. A system as recited in claim 26, wherein the coprocessor is located at the site of the proxy.

28. A system as recited in claim 26, wherein the coprocessor performs functions to guarantee that an application embedded in the coprocessor performs to a degree of security proscribed by the client and/or server.

29. A system as recited in claim 26, wherein the coprocessor functions to assure that the proxy can not tamper with the functioning of the agent.

30. A system as recited in claim 26, wherein the application embedded in the coprocessor adapts content supplied by the server to fit constraints of the client and the connection links.

31. A system as recited in claim 26, wherein the application embedded in the coprocessor adapts content supplied by the client to fit constraints of the server and the connection links.

32. A system for providing network security to a network employing a proxy, the system comprising:

a secure coprocessor located at the site of a proxy; and

a trusted application embedded in the coprocessor wherein the coprocessor delegates the task of enforcing an arbitrary trust model to the application.

33. A system as recited in claim 32, wherein the coprocessor functions to guarantee that the application is trusted to enforce the trust model between at least one server and a plurality of clients.

34. A system as recited in claim 33, where the coprocessor functions to assure the tamper resistance of the application.

35. An article of manufacture comprising a computer usable medium having computer readable program code means embodied therein for achieving client to server end to end security guarantees, the computer readable program code means in said article of manufacture comprising computer readable program code means for causing a computer to effect:

employing a proxy between the client and the server to provide connection links between the client and the server;

embedding a secure coprocessor for use as an agent of the client and/or the server;

the coprocessor acting as a converter between at least one protocol the client supports and at least one other protocol supported by the server; and

employing respective security protocols of said at least one protocol and said at least one other protocol.

36. An article of manufacture as recited in claim 35, the computer readable program code means in said article of manufacture further comprising computer readable program code means for causing a computer to effect the coprocessor guaranteeing that an application embedded in the coprocessor performs to a degree of security proscribed by the client and/or server.

37. An article of manufacture as recited in claim 35, the computer readable program code means in said article of manufacture further comprising computer readable program code means for causing a computer to effect the coprocessor assuring that the proxy can not tamper with the functioning of the agent.

38. An article of manufacture as recited in claim 35, the computer readable program code means in said article of manufacture further comprising computer readable program code means for causing a computer to effect the coprocessor adapting content supplied by the

server to fit constraints of the client and/or the connection links.

39. An article of manufacture as recited in claim 35, the computer readable program code means in said article of manufacture further comprising computer readable program code means for causing a computer to effect the step of adapting content supplied by the client to fit constraints of the server and the connection links.

40. A computer program product comprising a computer usable medium having computer readable program code means embodied therein for providing secure communication on a network, the computer readable program code means in said computer program product comprising computer readable program code means for causing a computer to effect:

securely embedding an agent at a proxy in the network, and

splicing a plurality of secure communication protocols of different protocol suites into the agent.

41. A computer program product as recited in claim 40, wherein the step of splicing includes splicing a security protocol of a Wireless Applications Protocol suite (WAP) to that of the Internet protocol (IP) suite.

42. A computer program product as recited in claim 40, wherein the splicing includes maintaining end to end security guarantees without a modification at the server.

43. A computer program product as recited in claim 40, the computer readable program code means in said computer program product further comprising computer readable program code means for causing a computer to effect the step of the agent performing at least one content adaptation function.

44. A computer program product as recited in claim 40, the computer readable program code means in said computer program product further comprising computer readable program code means for causing a computer to effect the step of maintaining a state of a splicing process resulting from the step of splicing.

45. A computer program product as recited in claim 44, wherein the step of maintaining includes employing a storage device external to the proxy, and using cryptographic means to encrypt the state.

46. A computer program product comprising a computer usable medium having computer readable program code means embodied therein for providing network security to a network employing a proxy, the computer readable program code means in said computer program product

comprising computer readable program code means for causing a computer to effect the steps of:

embedding a trusted application in a secure coprocessor located at the site of a proxy; and

delegating to a networking infrastructure a task of enforcing a trust model.

47. A computer program product as recited in claim 46, the computer readable program code means in said computer program product further comprising computer readable program code means for causing a computer to effect the step of guaranteeing that the application is trusted to enforce the trust model between at least one server and a plurality of clients.

48. A computer program product as recited in claim 46, the computer readable program code means in said computer program product further comprising computer readable program code means for causing a computer to effect the step of assuring the tamper resistance of the application.

49. A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for secure communication between a client and a server employing an untrusted proxy, said method steps comprising:

embedding a coprocessor at the proxy;

the proxy receiving a specific communication request from a client;

the proxy forming an n-tuple for the specific communication;

the proxy forwarding the n-tuple to the coprocessor;

the coprocessor generating a response, including a directive, to the n-tuple; and

the coprocessor sending the response to the proxy; and the proxy implementing the directive.

50. A program storage device readable by machine as recited in claim 49, wherein the coprocessor is a secure coprocessor.

51. A program storage device readable by machine as recited in claim 49, wherein the step of receiving includes:

awaiting a connection request from a first client;

creating an entry in a storage module for the client;

determining a sender of each received packet; and

retrieving a stored entry.

52. A program storage device readable by machine as recited in claim 49, wherein the n-tuple includes a sender id, an entry from a storage module and the received packet.

53. A program storage device readable by machine as recited in claim 49, wherein the client and the server can be either a sender or a receiver, and the step of generating includes employing a first protocol from the sender to the proxy and a second protocol from the proxy to the receiver and translating between the first and second protocols.

54. A program storage device readable by machine as recited in claim 49, wherein the translating includes decrypting the received packet as specified by the security parameters negotiated as per the first protocol and encrypting the decrypted packet as specified by the security parameters of the second protocol.

55. A program storage device readable by machine as recited in claim 53, wherein the translating includes modifying the received packet to meet constraints of the receiver and wherein the directive includes

forwarding to the receiver the packet resulting from the step of modifying.

56. A program storage device readable by machine as recited in claim 55, said method steps further comprising the step of aggregating a plurality of packets into a group of packets and performing content adaptation on the group of packets.

57. A program storage device readable by machine as recited in claim 49, wherein the communication between the client and the proxy employ protocols specified by the Wireless Application Protocol.

58. A method as recited in claim 1, further comprising the step of the coprocessor adapting content supplied by the server to fit constraints of the client and/or the connection links.

59. A method as recited in claim 7, wherein the splicing includes maintaining end to end security guarantees without a modification to a server involved in the communications.